

AD A052147

12

ARI TECHNICAL REPORT  
TR-77-A27

## THE FEASIBILITY OF ADDING GRAPHICS AND TEAM TRAINING SUPPORT TO PLANIT

by

Charles H. Frye

NORTHWEST REGIONAL EDUCATIONAL LABORATORY  
PORTLAND, OREGON

DECEMBER 1977

Contract DAHC 19-76-C-0008

Prepared for



U.S. ARMY RESEARCH INSTITUTE  
for the BEHAVIORAL and SOCIAL SCIENCES  
5001 Eisenhower Avenue  
Alexandria, Virginia 22333

DDC  
RECEIVED  
APR 3 1978  
F

# U. S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the  
Deputy Chief of Staff for Personnel

J. E. UHLANER  
Technical Director

W. C. MAUS  
COL, GS  
Commander

---

Research accomplished  
under contract to the Department of the Army

Northwest Regional Educational Laboratory

## NOTICES

**DISTRIBUTION:** Primary distribution of this report has been made by ARI. Please address correspondence concerning distribution of reports to: U. S. Army Research Institute for the Behavioral and Social Sciences, ATTN: PERI-P, 5001 Eisenhower Avenue, Alexandria, Virginia 22333.

**FINAL DISPOSITION:** This report may be destroyed when it is no longer needed. Please do not return it to the U. S. Army Research Institute for the Behavioral and Social Sciences.

**NOTE:** The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) ARI TR-77-A27	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) THE FEASIBILITY OF ADDING GRAPHICS AND TEAM TRAINING SUPPORT TO PLANIT.	5. TYPE OF REPORT & PERIOD COVERED (9) Rept. Final Jul 75 - May 76	
7. AUTHOR(s) (10) Charles H. Frye	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Northwest Regional Educational Laboratory Portland, Oregon	8. CONTRACT OR GRANT NUMBER(s) (15) DAHC 19-76-C-0008 new	
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Research Institute for the Behavioral and Social Sciences (PERI-OK) 5001 Eisenhower Ave., Alexandria, Virginia 22333	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (16) 2Q762722A764 I-764-B-2	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE (17) Dec 1977	
	13. NUMBER OF PAGES (12) 91 p	
	15. SECURITY CLASS. (of this report) Unclassified	
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES  Research was monitored technically by personnel of the Educational Technology and Training Simulation Technical Area of ARI.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  CAI, Team Training, Graphics, PLANIT, Computerized Team Training, Crew Training, Authoring Languages, Small Group Training		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This final report discusses the feasibility of adding a graphics and a team training capability to the ICU/PLANIT system.  ✓ Graphics capabilities which might be suitable for PLANIT are considered on three levels: simple line and curve plotting using a presently available, newly added function; line and curve plotting with magnification and limited animation using a new set of language directives which could be added; and, — Tom		

20. (continued)

the third level, line and curve plotting, animation, magnification, positioned text, and positional (e.g., light pen) response. The third level would require the new language directives and a new "Graphics" frame type. ↑

The first level is available now, subject to some added installation work. Level two requires only modest modification to PLANIT. Level three would require extensive modification efforts.

Similarly, the team training modifications are presented in differentiated levels according to the amount of system capability to be provided. The first level makes use of an improved DIAL capability and a "Common Lesson Matrix" both of which are now part of PLANIT. Additional levels of team training system support are not as well defined as for graphics and would depend on a further needs study.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



## FOREWORD

---

The Educational Technology and Training Simulation Technical Area of the Army Research Institute for the Behavioral and Social Sciences (ARI) conducts research to support the development of training concepts and evaluation techniques for applying automation, simulation and training devices in a unit setting. A training concept currently under study is the use of automation, viz., tactical computers, for training. Tactical computers have great potential for presenting individual and collective (or team) training. Individual training using the unmodified tactical computer and a machine transportable instructional software system called PLANIT (Programming Language for Interactive Teaching) has been developed and evaluated. The issue of collective or team training, as recently addressed in a Defense Science Board Report to the Director of Defense Research and Engineering, can now be addressed principally as a result of the work reported herein. In part, this Technical Report discusses the efforts to develop a PLANIT-based team training capability. Also presented in a discussion of the requirements for adding a graphics capability to PLANIT.

The research reported herein was sponsored by ARI and is responsive to specific requirements of the US Army Field Artillery School, the Training Support Center of the US Army Training and Doctrine Command, and to Army Project 2Q762722A764. The work reported on here was performed by the Northwest Regional Educational Laboratory under the technical monitorship of James D. Baker, Chief of the Educational Technology and Training Simulation Technical Area, ARI.

  
J. E. UHLANER,  
Technical Director

ACCESSION for		
NTIS	White Section	<input checked="checked" type="checkbox"/>
DDC	B II Section	<input type="checkbox"/>
UNANNOUNCED		<input type="checkbox"/>
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL.	SP. CIAL.
A		

## TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION . . . . .	1
CONDUCT OF THE STUDY . . . . .	6
GRAPHICS OPERATIONALLY DEFINED . . . . .	10
RECOMMENDED GRAPHICS LANGUAGE . . . . .	14
CONCEPTUAL GRAPHIC PROBLEM ADDRESSED . . . . .	21
MAINTAINING PORTABILITY WITH GRAPHICS . . . . .	31
GRAPHICS IMPLEMENTATION OPTIONS . . . . .	36
TEAM TRAINING DEFINED . . . . .	41
CONCEPTUAL TEAM TRAINING PROBLEMS ADDRESSED . . . . .	44
SUGGESTIONS FOR USING THE TEAM TRAINING DIRECTIVES . .	49
CONCLUSION . . . . .	57
FOOTNOTES . . . . .	59

APPENDIX A: CONSULTANT REPORTS

APPENDIX B: CONSULTANT CONFERENCE AGENDA

APPENDIX C: PRINTER GRAPHIC

FINAL REPORT: THE FEASIBILITY OF ADDING GRAPHICS  
AND TEAM TRAINING SYSTEM SUPPORT TO PLANIT

INTRODUCTION

PLANIT (Programming Language for Interactive Teaching) is a computer time-sharing system which contains the necessary software support for the authoring, editing and management of extensive training scenarios together with facilities for administering these to trainees and automatically recording data on their performance. PLANIT is among the class of software which is used for computer-assisted instruction.

PLANIT is also portable. Portability was a prime objective of that development effort which was begun in 1968 under contract to the National Science Foundation. The authoring and lesson execution facilities of PLANIT are competitive with similar systems but its portability makes it unique. Descriptive material on PLANIT, its language features, portability, etc., are available elsewhere and will not be included here.<sup>1</sup>

PLANIT's portability also influenced its adoption as the training subsystem component in selected tactical data systems such as DEVTOS and TACFIRE. Preliminary to the choosing of a training system for DEVTOS in connection

with the MASSTER Test 122, twenty-three training systems were analyzed for their applicability, resulting in the choice of PLANIT as the only one that met all the requirements.<sup>2</sup> Subsequently, PLANIT was installed on the DEVTOS system and used for MASSTER Test 122, the results of which were several recommendations for the continuance of such training on tactical data systems involving additional installations of PLANIT.<sup>3</sup>

In 1974, the U. S. Army Research Institute contracted with this Laboratory to assist with the installation of PLANIT on the TACFIRE computer, ANGYK-12. Unlike the commercial equipment (CDC 3300) used in DEVTOS, the TACFIRE computer was built to military specifications and lacked some of the support features which had been important to the installation of PLANIT, such as a FORTRAN compiler, floating point arithmetic, etc. Since it was important not to compromise PLANIT's portability, one of the stipulations of the contract was that the source code could not be changed to effect the installation. Instead, a FORTRAN-to-TACPOL translator program was developed which made the necessary alterations in the code automatically (TACPOL being the language of the TACFIRE computer). With the success of that installation effort, ARI clearly validated PLANIT's portability by running it virtually unchanged on a machine which was substantially different from any on which it had been run before. In addition,



the translator program which was developed to accomplish the installation task is sufficiently general that it can similarly translate any future versions of PLANIT so that improved versions can be installed at a tiny fraction of the original installation cost.<sup>4</sup>

In addition to the two tactical data systems already mentioned, ARI also implemented PLANIT on a UNIVAC 1108 at Edgewood, Maryland, and on a CDC 6400 at Fort Leavenworth, Kansas. By now, PLANIT has been demonstrated at Fort Hood, Texas, Fort Leavenworth, Kansas, Fort Monroe, Virginia and Fort Sill, Oklahoma. ARI has also demonstrated the compatibility of courseware (lessons) between sites, preparing materials on commercial systems which run on tactical data systems, and transporting the entire system, complete with training materials, from one tactical data system configuration to another. For a more complete description of ARI's installation experiences with PLANIT, the reader is referred to a paper written by Cecil D. Johnson of ARI and presented at the 1975 Annual Convention of the Association for Educational Data Systems, entitled: "Implementation of PLANIT at the U. S. Army Research Institute for the Behavioral Sciences."

Having established their interest in using PLANIT through these several demonstrations, ARI's next concern was the capability for the full utilization by the PLANIT training subsystem of each of the communications functions

that comprise the tactical data system operators' tasks. These include plotting lines and graphs, lighting illuminated buttons, sensing switch and button actions, training team members in a team setting, etc., this in addition to the usual instructional dialog that typifies the usual training system. The intent is to reproduce the operating environment in the training setting as nearly as possible to facilitate transfer of learning to the actual task. However, to accomodate these highly specialized communication task requirements without compromising the portability of the PLANIT system required some further study, for if PLANIT's portability was sacrificed for the sake of some specialized needs on a certain tactical data system, one of the most compelling reasons for its initial selection would be lost.

Therefore, in July, 1975 a contract was awarded to this Laboratory to study the feasibility of adding a graphics capability to PLANIT which would retain PLANIT's full portability characteristics. Also, in conjunction with another project, the feasibility of adding a team training component to PLANIT was considered. These two efforts merged, benefiting from joint consideration by an excellent team of consultants, and later, augmented by a small additional award, some experimental capabilities were added to PLANIT so that the needs in these areas of capability can be determined empirically before large sums

of development monies are committed. The remainder of this document reports the outcome of that feasibility study and describes the newly added PLANIT features. In addition, a new user's document has been prepared that includes descriptions of these new features.<sup>5</sup>

## CONDUCT OF THE STUDY

A series of seven major events highlighted the progress of the seven-month feasibility study:

1. Materials Acquired. Materials were collected for a variety of display devices, especially those which represented the several major kinds. These included, for example, California Computer Company's (CALCOMP) Plotter, the Tektronix storage tube displays, the PLATO Plasma terminal, the TICCIT television-linked terminal, as well as several of the BEEHIVE and ADDS type. In addition, specifications for the ANGYK-12 displays were acquired. The study of these documents provided the necessary background for the analysis of graphic display needs which are discussed later.

Additional materials were acquired pertaining to the team training aspect. These included the ARPA-funded LIS language manual from UCLA, and the PLANET (renamed from FORUM) computer conferencing system manuals from the Institute for the Future. Neither of these are, strictly speaking, team training systems but were the closest thing to it that could be found.

2. Consultant Conference. A panel of three outstanding consultants joined the principal investigator for a two-day working conference. The consultants received materials ahead of time so that they came prepared in the topics which



were to be discussed. Two of the three wrote their recommendations after the conference, giving direction to the remainder of the feasibility study and the recommended course of action for the future. Appendix A contains the agenda for the consultant conference and Appendix B contains the two reports. The three consultants were as follows:

Dr. Leon Nawrocki is from the U. S. Army Research Institute. Having desired that one member be from ARI, the choice could hardly have been better since Dr. Nawrocki is in the process of the most complete survey of literature on the use of graphics for instruction to date.

Dr. Harry S. Silberman, a professor from UCLA, was selected to represent the educator's viewpoint, which he is well qualified to do having been an Associate Commissioner for Education in the U. S. Office of Education and having held numerous teaching posts in addition to his current UCLA position. He is also thoroughly acquainted with training needs for large man/machine systems and the human factors involved in the displays, having headed training efforts for the SAGE (Semi-Automatic Ground Environment) air defense system during prior employment at System Development Corporation.

Major Larry Walker is an instructor at the U. S. Command and General Staff College at Fort Leavenworth, Kansas. He is thoroughly versed in the display and training needs for large data systems and is himself a very knowledgeable user of computing systems.

It should be noted that the consultants addressed themselves both to the graphics feasibility question and to the team training needs. In addition to their reports in Appendix B, their views form a framework for the remainder of this document.

### 3. Consolidation of the Views from the Consultant Conference.

A single, preliminary plan was devised for adding the features to the PLANIT system which were called for in the consultant conference. This entailed the adaptation of the recommended list of desirable features so that the new language directives would be structurally and syntactically compatible with the PLANIT system.

4. Washington Debriefing. A debriefing was held in the ARI headquarters on October 22 and 23, 1975 in which the preliminary plan (from point 3, above) was presented to selected staff members. The plan met with generally good acceptance.

### 5. Devise a Functional Plan for Integrating the New Features.

This step pertains to a detailed consideration of the coding requirements and interface parameters. This includes assessing the impact of adding each new language feature to the existing system, recognizing that some additions cause more programming havoc than others because of interdependencies with existing parts of the system.

6. Add Minimal Experimental Capabilities to PLANIT. One point of consensus in the consultant conference was that some of the new features being discussed needed more study before money should be invested on unproven new features. For a minimal additional investment, four new language features were added to PLANIT (FETCH, PUT, TERMINAL and SPECIAL) which will permit experimentation both with graphics and team training scenarios. Although the new possibilities are limited, they are believed to provide the necessary tools to determine which of the new language features are sufficiently useful to warrant making the modifications. They will also help to define the kinds of features which would enhance team training. Apart from their experimental value, these new additions will have continuing value to current PLANIT authors.

7. Document the Plan. This report is that document. It will discuss graphics and team training as separate components.

In regard to graphics, some basic concepts will be discussed, alternative configurations identified, and implementation options presented.

Since the options for the team training component still seem somewhat elusive, there will be discussion regarding the use of the new experimental features to empirically derive the needed language directives which would give the authors of team training scenarios what they need in the way of language aids.

### GRAPHICS OPERATIONALLY DEFINED

A graphic normally refers to any two-dimensional representation of line and/or pictorial information which is intended to convey information by its form, as contrasted to verbal information which attempts to communicate by word meanings. Most people have an intuitive sense of what to expect from a graphic. They might see a picture as "being worth a thousand words." While the above concept is quite reasonable for graphics in the general sense, it is not entirely true for computer graphics, and very different from the operational definition that has been found necessary for this study.

For the purposes of this study, a graphic will consist of any two-dimensional display composed of text, lines (straight or curved), and/or points which are so placed by reference to their intended position on the finished composite.

This definition was chosen to differentiate vector-generating full-screen displays from the clever kinds of things that can be done with output terminals that print one line at a time and eject between lines. Note that there exist a large number of cathode ray tube-type terminals which give the appearance of full-screen displays but in reality are line printing devices which scroll



upward as they display, much like a mechanical impact printer on paper. These latter devices, when used in the described manner, are not included in the class of graphic display devices. The operational definition given for a graphic display also does not include a few devices which, by montaging a projected film image or TV scan with the computer-generated output, create a full gray-scale picture. This latter is a function of the particular hardware rather than a component of the software aspects of the graphic language directives, so will not be considered.

Therefore, for purposes of clarity, this document will choose two representative output devices: a (mechanical, impact) printer and a (two-dimensional) plotter to represent the two classes of non-graphic and graphic devices, respectively.

The question can be (and has often been) asked, "Can PLANIT handle graphics?" The answer is clearly both "Yes" and "No." A clever author can cause something with the appearance of a graphic to result from line-by-line printing. Appendix C shows a sample of such a graphic which, when viewed from a short distance, portrays the picture quality of a studio photograph. PLANIT can do such things as this and even contains special language directives to assist the author in producing them. However, this is not a graphic according to the above operational definition since it is produced on a printer through a scrolling or

eject action. It makes no difference to the definition what character forms are used in the printing process.

On the other hand, it is easily conceivable that a computer program could direct a CALCOMP plotter to produce a full sheet of text, perhaps Lincoln's Gettysburg Address. PLANIT cannot now do this since it involves the deliberate positioning of the output (whether text, lines, dots, etc.) on a two-dimensional surface.

Therefore, the real question of interest in the graphics part of the feasibility study is not whether PLANIT can be made to produce a graphic such as the one in Appendix C but whether PLANIT can and should be made to position text, lines and dots on a two-dimensional surface by designating their target position along with the informational content of the output.

A second part of the graphics feasibility question is in regard to input rather than output, should PLANIT be able to sense a positional input such that the author can prescribe differentiated learning paths depending upon the position that was detected? This kind of positional detection might come through any of several different hardware devices, such as a light pen, a touch-sensitive infra-red beam triggering, a capacitively coupled pen, a "joy stick," a "mouse," or any of several other devices now on, or coming to, the market. For the purposes of this report, the light pen will be used to designate this class

of positional input devices. In order to complete the comparison between positional output and input devices, whereas the output (plotter-type) device communicates some kind of data to the designated display position, the input (light pen-type) device typically only communicates the position. In those few applications where some button action accompanies the position detection, this can most easily be handled as an independent input, and the two can be associated in the logic of the software. Therefore, for the purposes of this study, the light pen class of input devices will communicate only the screen position (in the form of an ordered pair of numbers representing the X-Y coordinates of the designated position).

Finally, if there should be any remaining doubt, the subject of this feasibility study concerns itself with the possibility of adding capabilities for interfacing to the plotter- and light pen-type of devices; no need has been expressed for enhancements to the printer interface.

### RECOMMENDED GRAPHICS LANGUAGE

The recommended additions to the PLANIT language to implement graphics fall into two categories, CALC commands and a Graphics frame. First, the CALC commands will be listed:

1. ORIGIN - Fix the intersection of the X-Y axes at any position on the screen.
2. LINE - Draw a line between any two specified coordinate points in relation to the origin.
3. CURVE - Draw a curve between any two specified coordinate points in relation to the origin for a specified radius, always assuming that the parameters are interpreted such that the direction of rotation is clockwise.
4. ERASE - Erase the entire accumulated display.
5. ZONE - Calibrate the X and Y dimensions of the display surface into the designated number of divisions of equal size.
6. HOLD - Hold the display steady on the screen for a specified number of seconds or minutes before proceeding to the next set of graphic directives.

The above six graphics directives will be compatible with the present CALC syntax for non-combinable functions. As such, they can have numerical arguments which can be supplied in the form of any legal CALC expression which resolves to a single number.



The number and description of the arguments for each of the functions is fairly obvious from the operation performed. ORIGIN will have two arguments, the X any Y coordinate values for the origin of the display. LINE will have four arguments, the two ordered pair sets of coordinate values which mark the endpoints of the line. Optionally, two arguments could be interpreted as the new destination to which a previously drawn line should be extended. CURVE arguments would be the same as for LINE but with the additional radius value argument. ERASE needs no arguments, being used simply to blank the screen. ZONE would have two arguments to designate the display divisions along each axis. More will be said about this function later. HOLD would have one argument, the number of seconds (or, optionally, minutes) to hold the display steady on the screen.

LINE and CURVE would have one optional additional argument that would designate the color of that portion of the display. If omitted, a default color would be chosen (recognizing that most displays are currently only one color anyway but the multi-color display potential will nevertheless be available).

The ZONE concept can be understood in terms of visualizing a matrix of lines overlaid on the display (tic-tac-toe fashion). The horizontal divisions (columns) will be designated by the first ZONE argument such that the value

of the argument would designate a corresponding number of columns at equal intervals across the display. The second argument would similarly designate the number of rows along the vertical axis, dividing the vertical dimension of the display into equal intervals. Although these row and column "lines" will not actually be drawn on the display, they will conceptually name the cell divisions (zones) of that matrix such that each rectangular cell can be referenced by an ordered pair of X-Y coordinate numbers, where the numbers refer to the geometric centers of each rectangle when used as parameters for any PLANIT language directives which must refer to a position on the display. The ZONE concept is occasioned by the need for transportability which will be discussed in more detail later.

The above set of language directives will be sufficient to draw a wide variety of line graphics. In order to organize the use of these directives most efficiently and to include verbal labelling of the picture and also a light pen-type of response, a new PLANIT frame type is being recommended, called a Graphic frame (or, conforming to the other PLANIT frame types, a G frame).

The G frame will be divided into five groups where more than one out of Groups two, three, four or five, are optional. This is consistent with PLANIT conventions for existing frame types.

Group one of the G frame will contain the frame number and an optional frame label, as in the existing PLANIT frame types.

Group two will consist of text to be displayed to the student together with the CALC statements which contain the above seven directives. The text lines will resemble text lines from Group two of a Q frame except that each will be prefaced with the coordinate arguments which will be separated from the text by a semi-colon (;). Two such arguments will be sufficient but a third, if present, would prescribe the color. For example:

12,4;START THIS TEXT AT ZONE (12,4) ON THE SCREEN

12,4,2;THIS TEXT AT ZONE (12,4) IS TO BE BLUE

The local site will determine the color-number assignments.

Group two lines which do not have the semi-colon in the above order will be taken as CALC lines. These CALC lines will follow the same general rules as for PLANIT's Programming frame. Any Group two line may have a line label at the start where the label is separated from the remainder of the line by a colon (:), just as in the P frame. Text lines may also have line labels.

Therefore, Group two of a G frame will produce the complete graphic display. In fact, it could produce several such displays separated by HOLD directives so the student would have a chance to view the display before the next one appears. Because of this need to HOLD the display, the animation effect can be produced by using short HOLD durations. Magnification and reduction of the display can result from changing the ZONE values and replaying the

Group two lines. Similarly, a change in the ORIGIN values would shift the picture (right or left, up or down) if the Group two lines were replayed with the new values.

If the Graphics frame is added to PLANIT, then there would be good reason to limit the use of the graphics directives to Group two of that frame.

Group three of the graphics frame would designate matching information for a light pen-type of response in a format similar to Q frame Group three lines. The format would be the answer tag (digits 1 through 9 since all answers would be numerical by definition), followed by an optional plus (+) correct answer designator. Continuing to the right on the Group three line, the format would now deviate from the Q frame Group three line to reflect the values that are expected to be in the response. Note that the response will always be an ordered pair of numbers reflecting the X-Y zone identification of the area to which the student pointed. Therefore, the Group three line will also contain an ordered pair of numbers for comparison purposes. this pair of numbers will be separated by a semi-colon (;) as follows:

1+ 5;7

It could be given in the form of expressions:

1+ X+4;3\*Y



Ranges of acceptable position numbers would be designated by separating commas (,):

1+ 3,5;9,12

The above would find a response in a collection of adjacent zones.

Matches and unanticipated answers would have the same interpretation as they now do in a Q frame.

Group four of the G frame would be identical to Group four of a Q frame. If any text is printed from that group (such as a feedback message), it would go to the printer device, not the display.

Group five is constructed identically to Group two of a Q frame, an arbitrary number of lines of text destined for the printer device. Group five would be a bypass provision which would execute only on installations which had no display provisions. Thus, for any given installation, Groups two through four of a G frame would execute, or Group five, but not both. Which executes would depend on a generation parameter in conjunction with whether a display device is available. The purpose of Group five is to refer the student to adjunct material if no display device is available at that installation.

This completes the language extensions needed for the graphic input and output capability. It should be noted that this does not necessarily constitute a single

indivisible entity that is being recommended but rather the above set of directives will be broken down at a later time into a set of options, graded according to implementation difficulty and expense.

## CONCEPTUAL GRAPHIC PROBLEM ADDRESSED

Why did seven months and several thousand dollars have to be spent on a feasibility study for using graphics when many excellent computer graphic systems already exist which even contain capabilities well beyond those being proposed here? There is a good chance that a question something like that might have crossed the reader's mind; it did for the principal investigator several months ago.

The answer is quite simple, though not always obvious: suitable machine transportable graphics packages are not known to exist. If PLANIT is not to compromise its portability, then currently existing graphics hardware/software packages must not be used.

Some work can be cited relating to machine independent applications involving graphics. Lyle Smith has documented a machine independent graphics subroutine package which essentially consist of a menu of display "inking" movements comparable to those being recommended here.<sup>6</sup> His implementation required that the local site translate the interface parameters into appropriate device movements just as PLANIT now requires for each peripheral device. This will be a PLANIT requirement for the display device as well. However, the directives for his subroutine packages were

designed for inclusion in FORTRAN programs, being particularly appropriate for graphing mathematical functions so that the benefit to PLANIT did not extend much beyond the principal used to achieve the interface.

The CALCOMP Plotter documentation is also of interest because this device has been wired to so many different computers that it has legitimate claim to machine independence. Again the "inking" directives are of particular interest and these correspond fairly well with those being proposed. The Plotter adds such directives as "Pen up" and "Pen down" which is a function of its mechanical action. These directives can be implied in the set being proposed for PLANIT. The interface offers little help, being a combination of hardware and software which is designed specifically for the computer to which it is being wired.

Before the transportability problems could be identified with which this study would have to deal, the nature of the desired graphic displays had to be identified. This problem was addressed by the consultants. What emerged was the "blackboard model" which meant that the author should be allowed to use the display surface in a fashion at least comparable to a lecturer using a chalkboard.

The blackboard model provided an excellent frame of reference for devising the list of desirable language directives. It suggested the need for a good line drawing



capability. If axes are displayed, the location of the point of origin must be optional. There must be a capability for labelling the figure.

Hardware conveniences and constraints have modified the model to some extent. Some examples follow:

1. Magnification/Reduction. This feature is very useful in computer-generated displays. It allows the viewer to observe a "close-up" for detail, or a distant perspective for pattern. It deviates somewhat from the blackboard model because of the physical difficulty of redrawing a geometrically similar figure with increased or reduced dimensions on the chalkboard. However, since the computer-generated display originates in a set of language directives, it is only necessary to replay that display-producing section of the program with a different size parameter to see the same display in a different size. In the case of the directives suggested here, the arguments for ZONE can be variable, taking on different values for subsequent replays of the frame which produces the display.

2. Animation. The consultants agreed that animation was probably not worth extra investment. There seemed to be no evidence that it was beneficial to learning. However, when successive displays are generated by the computer, because of its speed, some directive must be given to hold each figure steady long enough for the viewer to see it. Otherwise the display would pass by as little

more than a flash on the screen. Or, if the display happens to be drawn on paper, enough time must elapse for the viewer to provide a clean sheet of paper between figures. The HOLD is designed to serve this function. It has no direct analogy in the blackboard model. Since the HOLD directive is necessary, it only must accept small time intervals in its argument to provide the effect of animation. Thus, the animation feature comes as a bonus for no extra investment.

3. Selective Erase. The consultants agreed that the author should have the opportunity to selectively erase portions from his figure even as an instructor might selectively erase parts of a chalkboard figure. Selective erase directives were considered for a time and were even included in an earlier report to ARI. However, selective erase directives are not being recommended here because they violate the first premise that the PLANIT code must retain its portability. Since the portability issue is discussed in the next section, further comment will be deferred.

4. Terminal Configuration. Until now, the PLANIT effort has been so engaged in finding solutions to so many other transportability problems that alternatives in terminal equipment were kept to a minimum. Some things had to be considered such as character set (number of characters and their colation sequence), what to do with optional lower case inputs, etc. However, having allowed for those differences, the terminal has been assumed to be the

"printer" type, i.e. single lines of print that scroll upward as new lines are printed. Now that the addition of graphics is being considered, its effect on the nature of the terminal configuration is important. The main issue is whether to expect the printer function and the graphic display function to be in separate physical devices or consolidated in one device. There are samples of each:

A CALCOMP Plotter is normally an adjunct device and is most often a resource that is shared by all active users on the system who need to use it. The same is often true for large, vector-generating cathode-ray tube displays of the refresh type. In each case, the users often communicate through other terminal equipment and route their displays to one of these devices.

On the other hand, there are several terminals which are equipped to generate graphics on the display surface that the user views for his normal communication, i.e. the "printer" and "plotter" functions converge on the same display surface. The Tektronix 4000 series terminals, the PLATO Plasma terminal and several micro-processor-driven "smart" terminals fall into this category.

The difference between these two terminal configurations is a matter of great conceptual importance to PLANIT. Will there be a graphic display on the screen at the time some text is to be printed? If so, will the graphic figure scroll upward with the text? If so, this would make it difficult to hold a figure for viewing while discussion was

being carried on about the figure. If not, then the display device must permit selective erasure of the preceeding lines of text without disturbing the figure so that these lines can be scrolled upward to make room for the next line at the bottom. Some displays (such as the Tektronix 4000 series, mention above) do not permit this degree of selective erasure. The option of manipulating a representation of the screen in memory and repainting the screen after each change would be hopelessly cumbersome, especially when trying to resolve one image passing over another. Even the ability to selective erase does not solve the problem since intersecting display points will be erased leaving gaps in the graphic figure.

About the only reasonable solution that has been found to this terminal configuration problem is to include parameters in the PLANIT system generation process which indicate the characteristics and configuration of the terminal equipment, making PLANIT do different things with output accordingly. For example, if the "plotter" function is in a separate device from the "printer" function, PLANIT can continue communicating through the printer while the plotter display is being held or sequenced through a set of displays independently. If the two functions must occur in the same device, then the screen should probably be erased when switching from one function to the other. This solution is far from ideal since strategies for lesson preparation would necessarily be different for the



two options, introducing the likelihood of incompatibility of lesson materials from one installation to the next. Therefore, some PLANIT lessons will ultimately require certain terminal equipment configurations to run properly.

5. Designating Figure Positions on Screens of Unknown Dimensions. Assuming that lesson material may be prepared on a system with one kind of graphic display device and then run on another, some method other than absolute screen location parameters must be used since industry standards for these parameters do not exist. This is simply not a problem for a training system such as PLATO where identical terminal devices of a single type is assumed.

Since display screens characteristically have differing dimensions (from a few inches to several feet), differing resolution, some with multi-colors, a variety of character sets (some programmable), differing character generators (some with optional sizes), differing methods of generating lines, arcs and circles, etc., attempting to achieve a degree of uniformity in a displayed figure is a chore.

The "zone" concept was devised as a reasonable solution to several of these problems. The ZONE language directive with its two arguments, operationally defines the concept.

The unknown screen is only assumed to be square. (In cases where even that is not true, the non-conforming portion of the screen can be left dark). Not knowing the

actual resolution of the screen (i.e. the actual coordinate numbers to use to achieve proper positioning of his intended figure), the author conceptually divides the screen into zones. He determines the number of zones by the numbers he supplies to the ZONE function. ZONE(10,10) would indicate a ten-by-ten matrix, or 100 zones. The author may choose as fine or coarse a matrix as he wishes. The zone numbers along each of the X and Y axes will then be the calibration used by the author to position his figure.

When drawing lines and curves, the endpoints of the sketch will be the geometric center of the indicated zones.

When positioning text on the screen, text size will be chosen which most nearly evenly fills the cell size of the zone (that is, if a choice exists).

When indicating a screen position to which the user pointed his "light pen" device, the zone coordinates will be returned to PLANIT that bounded the spot, and the response matching will refer to these zone values.

The chosen number of zones does not need to agree with the absolute calibration of the screen; that conversion will be made automatically within PLANIT. Conversion formulas will be established from system generation parameters which convey the actual calibration values.

The value of the zone concept is that the author may refer to relative positions on the screen in units of his own choosing, and that displays which are created on screens

of one size will probably be equally usable on another of a different size. Also, for acknowledging "light pen" type of responses, the zones relate the light pen action to the figure which was just displayed, not to the actual characteristics of the device.

Finally, the ZONE parameters are just two values supplied with that directive. As was already mentioned, those values may be supplied in variable form so that the figure-generating frame can be replayed with new ZONE values, effectively magnifying or reducing the total displayed graphic.

6. The "Bypass" Group in the Graphic Frame. Group number five in the Graphic frame was designated for text which would appear only to the user who had no graphic device available. This would provide opportunity for authors to write lessons which would use the graphic capability if it existed, or refer to printed pictorial handouts if it didn't.

The bypass group was one of the recommendations from the consultants. It has a great deal of merit, especially in regard to an attempt to maintain compatibility of lesson materials even in the absense of important hardware.

The bypass group suggestion also has certain shortcomings which leads to some ambivalence about how strongly to recommend it.

First, it would be the only five-group frame in PLANIT. This would not only break the current pattern but would also

add to the programming difficulties for including it in the PLANIT system since only two- and four-group frame structures are now legal.

Secondly, without Group five, the other four groups are very analagous to the present Question and Programming frames. Group five would constitute a deviation from the pattern, increasing learning difficulty for authors by some small margin.

Thirdly, filling out all five groups would guarantee some waste in lesson storage space since information for two kinds of systems would be maintained while only part of it would be usable for any given system.

Finally, there is good reason to object to generating the kinds of graphics on a computer which could be printed on a piece of paper. If the less expensive medium is satisfactory, there is little reason to put it on the computer in the first place.



## MAINTAINING PORTABILITY WITH GRAPHICS

The many transportability problems among the typewriter class of terminals are only multiplied in the terminals which display graphics, among which differences abound. Methods of "painting" the display include: impact printing, thermal printing, inking with a pen, illuminating dots in a plasma panel, illuminating light-emitting diodes, illuminating a phosphorous coating in a storage tube, deflecting ionized ink streams, deflection of an electron beam in a vector generating CRT, synchronized control of the energy level of the beam in a full-screen raster sweep (TV raster), and probably several more. Numerous kinds of response units have already been cited on page 12.

Commonality is more difficult to find than differences. What commonality exists is probably at the level of the image that is perceived by the viewer.

One area of hardware commonality is that each of these graphic display devices has a discreet, numerical position address for each point of departure and termination on the display surface. This is true of devices which appear to produce continuous drawings as well because of the discreet nature of numbers in digital computers. What conversions take place between the computer and the

display device through such interface units as D-A converters is of no consequence to this discussion.

Because position addresses are discrete, there exists some calibration in all such devices such that the resolution of the display surface is a function of the minimum increment size of the screen calibration. This minimum increment is sometimes called the "bucket size" of the display.

Since this bucket size is a universal characteristic for graphic display devices, it can be supplied to the PLANIT system generation process in the form of a parameter. The parameter value can then serve as a constant in an equation where the ZONE value is variable, yielding absolute screen position values for display segments which are supplied by the author in ZONE units. This will guarantee that a figure reproduced on a different display from which it was created will appear in the same size and aspect ratio as it did on the originating screen. Differences in resolution could cause the apparent "smoothness" of a line to change. However, because of the structure of the ZONE concept, the change could be a greater refinement of the figure than was seen on the originating screen, suggesting a true machine-independence being obtained by this concept.

There are several characteristics of certain displays which can be implemented on other displays only with substantial difficulty, if at all. For example, the plasma panel in the PLATO terminal consists of a large dot matrix

where each dot can be selectively illuminated or extinguished. This has a lot in common with an LED (light-emitting diode) display but very little in common with a storage tube display where the image is painted on a phosphorous coating to remain until it is no longer visible because of its decay rate or until it disappears because the screen is uniformly flooded with light. It will also obviously be rather difficult to selectively erase an inked display.

Multi-color displays are still the exception rather than the rule and lack any display standards.

Character size for the internally generated character set is determined by the hardware of the device. There is sometimes a choice among two or three character sizes but authors will not be able to count on any consistency of character sizes from one display device to the next (other than what one would generally expect from a human factors point of view to make the text readable). Character sets also vary but PLANIT procedures already exist to neutralize that problem.

Probably the most serious problem is the one pertaining to configuration, discussed on page 24. If a training scenario is written for a configuration wherein the trainee can carry on a dialog with the computer while he is viewing the display, it will be extremely difficult to make that same lesson scenario work effectively in a setting where the textual and graphic output must share the same display surface.

Even maintaining separate devices may not solve the problem since it is common for facilities which operate configurations of this type to expect the users to share a lesser number of graphic displays, often spooling the output to the devices in order that output directives from different users will not get intermixed on the same display. If such a delay is built in to the production of the graphic, it would be virtually useless for PLANIT needs unless the author anticipated the delay in writing the lesson material.

The response units such as light pens pose fewer portability problems since nearly all are designed to convey only an ordered pair of coordinate values which identify the spot on the display at which the user pointed. The number values can be easily converted by the ZONE equations to fit within the scheme of the Graphics frame.

However, if it should be desired to use the light pen device to "draw" on the screen, the problem is complicated. Some of the devices such as the light pen will trigger only when it detects the light from a portion of the display. Therefore, to draw with a light pen, it is necessary to first display a dot matrix so the pen will have light sources to sense. This is not true for some of the other comparable devices such as a joy-stick, capacitatively-coupled pen or touch-sensitive grid where position information is detected independently from any display that happens to be on the screen.



A drawing capability is not being recommended at this time beyond that which an author could cleverly devise using the Graphics frame.

## GRAPHICS IMPLEMENTATION OPTIONS

The graphics package which has been outlined in this document is designed to provide PLANIT users with very flexible display capabilities. However, this is not an all-or-none package. At least three options are reasonable which provide a degree of graphics capability for less investment. The first option is in the use of the SPECIAL language directive.

A PLANIT directive called SPECIAL has been added to the CALC language. Although this addition was not a part of the statement of work, the benefits of it to the study will be obvious and it was possible to include it within the scope of the project. Details of its use are presented in the new user's document, "PLANIT LANGUAGE EXTENSIONS THROUGH VERSION 2.8."<sup>5</sup>

The SPECIAL function provides a means of transmitting a special call to the MIOP interface program for work not now included in the normal installation of PLANIT. No change needs to be made to the PLANIT code to implement this directive. It is already there. There is also provision to pass 10 integer values with the call so that different codes could be used to initiate many different activities. Provision is also made to pass a numerical

value back to PLANIT which, as far as CALC is concerned, becomes the resolved value of the function, comparable to the call of any other CALC function. Before SPECIAL was added, any new function to be added to PLANIT had to be coded in its source code and associated with some language directive. Now, using SPECIAL, the installer is permitted to define almost any new function he desires by deciphering agreed upon codes in MIOP, a subroutine that is written locally, therefore well understood.

Using SPECIAL, the local site can implement simple graphic operations such as the drawing of line and curve segments. This implementation of graphics will be subject to several restrictions but can be accomplished at very low cost. No further work needs to be done to PLANIT to put out line drawings or plotter graphs. Some text could be put on the screen as well although it would be somewhat clumsy. Unless only one user would be on the system at any one time, a HOLD operation should not be implemented since all users would have to wait for the one to view his screen. Therefore, the author would have to space his displays around question/answer frames so that the screen would remain steady until the user typed his response. Also, this option would assume a display device which was separate from the user's communication terminal. Merging his dialog and the figure on the same screen would be an unreasonably complex task for the MIOP programmer.

Use of these graphic commands can be made very easy for the author by defining appropriate plotting functions in CALC, each of which already contain the correct SPECIAL call sequence. Essentially, this could give the authors the capability of drawing lines and curves, inserting a few characters on the screen for labelling, and an erase.

The second option does require some work on the PLANIT source code but would take no more than a dozen or so weeks to complete. Essentially, this would provide each of the six named CALC directives: ORIGIN, LINE, CURVE, ERASE, ZONE and HOLD. Drawings produced by the LINE and CURVE directives could be moved about on the screen using the ORIGIN directive, or magnified or reduced, using the ZONE directive. HOLD would be implemented in such a way that only the one viewing the display would become inactive; the other terminals would continue working.

This second option also assumes that the graphic will be displayed on an adjunct device. No provision would exist for positioning text on the display screen.

The third option is essentially as recommended earlier with the six directives to be used in a new Graphics frame. This option would require a considerable amount of work on the PLANIT source code since the adding of a new frame type affects a substantial part of the system, from creating a lesson and editing it to execution and saving on tape, and would require from nine months to a year to complete.



Adding the fifth group to the Graphics frame would lengthen the job substantially since PLANIT now has no five-group frames so that group ranges would have to be modified throughout the program.

This third option would provide all the capabilities of the first two options plus several more. Text could be displayed anywhere on the screen. Light pen-type of responses could be collected and processed through PLANIT's normal lesson logic. All the output to the student could be consolidated on one screen if necessary, with the screen alternating between "printer" kinds of data and "plotter" (graphic) kind. However, having the capability to position text on the screen, authors could avoid data in the printer format if they so chose, creating displays which also contain all necessary communication.

All three options will require installation work to be done on the MIOP subroutine at the local site. The amount of this work would not vary appreciably among the options.

Having added this capability to PLANIT, the entire package would then be portable, using a wide variety of equipment in a virtually identical manner. Although the fifth group of the Graphics frame was recommended to further increase transportability, running a graphics lesson on a computer without a graphics capability, this is not seen as being a very practical transportability.

A lesson which effectively uses a graphic display simply will not have the same teaching properties if executed without the display equipment. If it does, then one should forget the expense of using the display and print pictures. Therefore, transportability will require comparable equipment, though not identical.

In summary, the graphics recommendations for PLANIT consists of six new language directives (in addition to the new "SPECIAL" directive that is now operable) and a new graphics frame. It is reasonable to consider the acquisition of graphics capability in PLANIT in terms of a set of three options:

1. Using the new SPECIAL PLANIT directive, line figures can be produced with no further change to the PLANIT code.
2. The six graphics language directives (ORIGIN, LINE, CURVE, ERASE, ZONE, HOLD) can be implemented and used in current CALC statement formats to produce figures which can be magnified, reduced, held or animated.
3. Implement a graphics frame to provide both a context for the six new directives (above) and a designated positional response. The recommendation favors a four-group graphic frame although a five-group frame could be implemented. This option would also provide the addition of text to the display.

Either of the three above options imply added installation work at the site to interpret the graphic codes in the MIOP program and produce the necessary segments of the display. Installation work for any of the three would be approximately the same.

### TEAM TRAINING DEFINED

Some problems are too large or decentralized (or both) to assign the solution to just one person even with all the assistance which computing machines makes available.

Tactical data systems provide a ready example. Even with sophisticated computers to support the operations, yet several duty stations are necessary, each having a vital role to play in the total system operation. Taken together, the people that occupy these duty stations form a team.

It is readily obvious that training facilities are now available to provide training for the members of the team with respect to the duty stations which they will occupy. The team member can even receive this training on the computer via PLANIT (or another comparable training system). However, the training will be about the tactical data system rather than in the tactical data system.

Though the difference is subtle, it is very real. While in the duty station position of the tactical data system, the operator will be interacting with other team members in the process of performing the task while in the individualized training mode the other members are absent.

In recognition of the need to provide training with other team members present, simulation exercises are

12

regularly scheduled in which the team operates the tactical data system with mocked up data. This helps to bring the "team" elements of the task into the training environment.

There is nothing wrong with the training value of these kinds of simulation exercises. The problem is with the economics of the situation.

While running a tactical data system in the simulation mode, a referee must be constantly present. The referee cannot be just any member of the team; he must be thoroughly versed in the entire operation of the system so that he can immediately detect problems at any of the duty stations. Such referees constitute a scarce resource that may not always be available at training sites, especially in critical need situations.

Also, simulation exercises are not ideally suited for novices. If all the duty stations are occupied by novices during the simulation training exercise, a fair amount of efficiency will be lost while the team members are "swapping ignorance." Yet, each of the novices must be brought into the team experience at some point. Quite often this will mean bringing a novice into a team where the other members will have already gained some amount of experience. Under such circumstances, studies show that the pace of the entire team will be held to that of the novice. In essence, an experienced team will devote valuable time to bring the performance of a novice up to some acceptable level of



proficiency. That is some teacher/pupil ratio! Yet it is apparently not too far removed from present practices.

The team training capability envisioned by this study provides an intermediate step between the individualized training about the system and the online simulation training exercises. This new system would bring the team members together in an automated environment where the members would be made to function as a team, even as in the simulation training, but the system would also retain its automated training propensities. No referee would be necessary since the training scenario would automatically monitor the performance at each duty station, interacting with that operator to correct any errors. The entire team could be made up of novices for the same reason since each duty station would be monitored simultaneously, and only correct performance would be passed into the system.

This automated team training system is not being suggested as a replacement for simulation training but rather as an intermediate step to bring up the proficiency of novices to the point where they can be functioning team members. The economy of providing this training to a team of novices without a referee present is apparent. It is this kind of team training facility that is envisioned here, a separate training system that would execute on the field-configured tactical data system hardware. And as a post script, a team training facility such as described here would seem to have application far beyond the tactical data system environment.

#### CONCEPTUAL TEAM TRAINING PROBLEMS ADDRESSED

The conceptual problems with regard to the team training facility are nearly opposite those of the graphics, mentioned earlier. Whereas in the case of graphics, the language structure was defined rather precisely with some confidence that the recommended set would accomplish the desired task, comparable language directives for team training are not so readily inferred. Also, portability considerations posed a major set of problems for the graphics implementation but not so for the team training; only one portability issue needs to be addressed for team training, and that one is not expected to constitute a major problem for most hardware configurations.

The fact that a graphics language structure was not too difficult to plan out is due largely to the fact that graphic devices are common to many military and commercial computer systems. Each already has a language for using the device. Therefore there already exists a good backlog of experience from which to draw. It is in the area of portability where the work is sparse. In the case of team training facilities, those that exist scarcely show any commonality, usually being designed with a single specific training situation in mind. Therefore, proven language

structures for a generalized team training facility are not known to exist. On the other hand, there is reason to believe that the team training, once the capability has been provided, will communicate through the normal channels for which portability problems have already been solved, so the question of portability with regard to the team training facility should not pose much, if any, additional difficulty.

Regarding the language structure for team training, it can be supposed that at least the following four options should be available and under the control of the training scenario author with respect to each trainee station:

1. Ability to carry on a normal automated training dialog with the computer in a fashion similar to that which PLANIT now provides.
2. Ability to review, correct and transmit an input from the trainee to any other team member, or to the system itself.
3. Ability to intercept, modify and/or relay any input from the system or other trainee for display to the designated trainee.
4. Ability to add to and update a common data base which reflects the current status of the training situation and/or the accumulated experiences of the trainees and to use selected subsets of this data to determine the next course of action in the training session.

It is reasonable to require that the above four options operate either in the synchronous or asynchronous mode, that is, that the options be allowed to occur either according to a predetermined, fixed sequence or spontaneously as the need arises. Differing training needs will suggest one mode or the other, or perhaps both in some prearranged mix.

Two systems were examined which specifically assume several simultaneous interacting members on the system: LIS and FORUM.

LIS/11 (Laboratory Implementation System, PDP-11 Compatible Version) is designed for controlling message flow between members of a team.<sup>7</sup> It is an ARPA supported project at the Center for Computer-Based Behavioral Studies, University of California at Los Angeles. LIS contains language directives for the review, editing and routing of messages to any individual or designated group on the system. The scenario prescribes several role levels for the players, which ones can monitor the activities of others, and the amount of control the monitor can exert. LIS also provides a facility for voting.

While the control of message flow would certainly be a vital component of a team training system, LIS apparently does not provide the means of dispensing training or for authoring a training scenario. While training is undoubtedly a part of the LIS system, it must come from other players. The team training facility envisioned for PLANIT can certainly profit from the message control procedures in LIS but the existing set of LIS language directives are not likely to be appropriate because of the need for an internally stored training scenario. Such a training scenario would be expected to substantially alter the method of message handling from that used by LIS.



The other system, FORUM, is designed for computer conferencing. FORUM was developed by the Institute for the Future under support from the National Science Foundation and ARPA,<sup>8</sup> and was a forerunner to a computer conferencing system now called PLANET which can be accessed through a commercial time-sharing service.<sup>9</sup>

FORUM, like LIS, is a message handling system. It is not intended for team training but rather for the orderly stacking, queuing and selective displaying of messages which originate from other members of the conference. Therefore, considering the difference in purpose for the FORUM system from that envisioned for team training in PLANIT, plus the structural differences of the two languages, about all that can be gained from the FORUM work are some ideas concerning the proper handling of inter-terminal dialogue.

In the absence of appropriate models to guide the process of defining a usable set of team training language directives, the consultant panel favored a set of experimental directives which can be used on a test basis to try some team training scenarios. Such experimentation would serve to define those language directives which were really needed by authors. Three new PLANIT language directives were added for this purpose: FETCH, PUT and TERMINAL. A change was also made to the DIAL directive. These will be discussed in detail in the next section, and reference manual information is also available.<sup>5</sup>

Only one new portability problem is of concern because of the team training option. This is primarily due to the asynchronous nature of the operation. Since messages may be exchanged at any time, they must get through to the terminal even if it is in the read status. If that is the case, the terminal must revert back to the read status again after the message has been delivered. The potential problem is seldom due to hardware. Software system service routines are often used in the coding of MIOP to implement terminal communications and these sometimes place restrictions on the read/write sequence. In that case, it may be necessary to forget those routines and code that part at the machine level. Without this provision, messages could back up in the buffers causing the lessons to suspend operation and log out the user.

## SUGGESTIONS FOR USING THE TEAM TRAINING DIRECTIVES

The team training directives which have been added to the PLANIT code provide three new features:

1. A common lesson matrix which can be defined, retrieved, or updated. The FETCH and PUT directives manipulate the common lesson matrix.
2. Use of the DIAL directive while in CALC or as a CALC command in a lesson scenario. In both cases the target terminal may be designated by a CALC item name whose value is the intended terminal number.
3. The directive, TERMINAL, is a CALC item name whose value is equal to the number of the user's terminal.

Regarding the common lesson matrix, it consists of a save area that will accomodate a standard CALC user-defined matrix of any dimensions. Having defined the matrix, the common area is then defined by the PUT x operation where "x" is the matrix name. FETCH x will retrieve the data for any user of that lesson so long as his matrix "x" has the same dimensions as the original (the matrix name may be different). The update operation consists of a FETCH followed later by a PUT with intervening CALC statements which update the working matrix. This entire sequence should be completed within one group of a PLANIT frame.

The common lesson matrix is valid for all trainees on a given lesson or chain of lessons. The common lesson matrix will be defined by the first lesson in the chain. Other lessons may also use a common lesson matrix but these will be distinct common areas and no data may be exchanged. A chain of lessons consists of those which automatically branch to the next lesson in the chain. The trainee's entry point will always be the first lesson in order to assure the use of the proper common area.

The common lesson matrix will be useful for such things as accumulating performance across trainees, providing a common data base, transmitting codes which specify desired activities for designated team members, implement synchronous or asynchronous operations as desired, transmitting CALC data from one chained lesson to another, etc. The common lesson matrix will operate differently from the LINK matrix in that the LINK values will be unique for each trainee whereas the common lesson matrix will be used by all trainees (on a given lesson chain).

By adding the DIAL directive to the CALC mode, the lesson and/or the trainee can easily direct a message to any other operating PLANIT terminal. This assumes that PLANIT is performing the time-sharing function for its terminals, as does the use of DIAL in the Command mode. The value of the variable item name terminal designator in the CALC use of DIAL will become more obvious in the examples. Using the TERMINAL directive, the lesson logic



can determine the other team members, assigning their terminal numbers to variable item names which are then used in the DIAL command, relieving the trainee of having to know the terminal numbers of the other participants.

The TERMINAL directive is most useful in the team training format, to provide the automatic capability of identifying one's terminal number to other members of the team. Otherwise, there is little reason to need to know the terminal number. It will be the value that is printed on the terminal at login time.

These new directives will enable the preparation of a team training scenario. There will be some limitations. Lesson logic will necessarily be more complicated than the usual individualized lessons. However, there will be sufficient flexibility to experiment with a wide range of team applications, which in turn should suggest some new language directives to ease the task for future authors.

The remainder of this section will present some examples of selected lesson operations which will be needed to implement various kinds of team training procedures.

First, it will be necessary for the lesson to know that the team is in place and ready to begin training. Following is an example of a two-person team acquisition logic:

FRAME 1.00 (D)

G2. CRITERIA

C:SET MATRIX(X,20) C:SET X(1)=TERMINAL C:PUT X  
F:@TYPE 'HI' TO CONNECT WITH THE OTHER PLAYER.

FRAME 2.00 (Q)

G3. ANSWERS

A HI

FRAME 3.00 (D)

G2. CRITERIA

C:FETCH X

IF X(1) EQ TERMINAL

F:NOT THERE YET. WAIT A MINUTE AND TRY AGAIN. B:2

ELSE C:SET HIM=X(1) C:X(1)=TERMINAL C:X(2)=TIME C:PUT X

C:PRINT 'OK, YOU ARE LINKED WITH TERMINAL ';HIM ROUND(0)

IF TERMINAL LS HIM C:SET MINE=3 C:SET HIS=4

ELSE C:SET MINE=4 C:SET HIS=3

In the above example, the two players will be connected to the team scenario by both GETting the same lesson name. Beginning with the third statement in frame three, the logic is set up such that the most recently answering terminal will be identified by number in the first common entry, and the time of the answer in the second entry. The item HIM will be the terminal number of the other player and can be used in the DIAL command, e.g.:

DIAL HIM YOU AND I ARE NOW TEAMED.

The third and fourth entries of the common area are set up for communicating code values, and the items, MINE and HIS are defined properly so that separate entries will be assigned. MINE and HIS would be used frequently to subscript the matrix X after a FETCH or between a FETCH/PUT

update sequence.

The next example will perform the initial acquisition for three team members. This logic can be extended to as many members as desired. Having acquired all members of the team, a branch is made to another lesson where each team member will be in a different lesson. The common lesson matrix will be valid for all three lessons. Each lesson will then proceed according to the role of that particular team member:

FRAME 1.00 (D.)

G2. CRITERIA

C:SET MATRIX(X,20)

IF LINK(10) NQ 0 C:PUT X C:LINK(10)=0

ELSE C:LINK(10)=1 C:FETCH X C:LINK(10)=0

IF X(I) NQ 0 FOR(I=1,3)

F:SORRY, ALL POSITIONS ARE TAKEN. ANOTHER TIME. C:FINISHED

ELSE F:@ARE YOU RED, YELLOW OR BLUE?

FRAME 2.00 (Q)

G3. ANSWERS

O KEYWORD ON

A RED

B YELLOW

C BLUE

G4. ACTIONS

A C:SET COLOR=1

B C:SET COLOR=2

C C:SET COLOR=3

- R:ANSWER ONE OF THE THREE OR TYPE 'FINISHED.'

FRAME 3.00 (D)

G2. CRITERIA

IF X(COLOR) NQ 0 F:SORRY, WE HAVE ONE OF THOSE ALREADY.

F:CHOOSE ANOTHER. B:2

ELSE C:FETCH X C:X(COLOR)=TERMINAL C:X(19)=TERMINAL

C:X(20)=TIME C:PUT X B:5

FRAME 4.00 (Q)

G2. TEXT

DON'T HAVE ALL THE PLAYERS YET. TYPE 'GO' AND I'LL  
CHECK AGAIN.

G3. ANSWERS

A GO

FRAME 5.00 (D)

G2. CRITERIA

C:FETCH X

IF PROD X(I) FOR(I=1,3) EQ 0 B:4

ELSE F:OK, LET'S GO. B:RED, YELLOW, BLUE; COLOR

In this example, the first entry of common has the terminal number of RED, the second has YELLOW, and the third has BLUE. The 19th entry shows the number of the most recent terminal to answer and the 20th entry shows the time of the answer. Finally, a branch is made to one of the three lessons, each of which presumably contains logic that pertains to a particular player.

There may be times where all players are to move through a sequence together. It may be important in another setting that no player begin a new topic until all players have completed the current one. Or it may be desirable to cause the players to take turns in responding. Each of these cases implies a synchronous form of operation. Both of the above lesson examples show instances of synchronous operations since no player is allowed to proceed until all have signed onto the system. The following is another example. In this case no one is allowed to proceed beyond frame 10 until all are together:



FRAME 8.00 (D)

G2. CRITERIA

C:FETCH X C:X(COLOR+10)=10 C:PUT X

F:THERE MAY BE A SHORT DELAY UNTIL EVERYONE CATCHES UP.

FRAME 9.00 (D)

C:FETCH X

IF X(I) EQ 10 FOR(I=11,13) B:11

FRAME 10.00 (Q)

G2. TEXT

TYPE 'GO'

G3. ANSWERS

A GO

G4. ACTIONS

F:OK, WILL CHECK B:9

If this were instead a synchronized questioning of the team members, another question frame could be inserted in the sequence.

Asynchronous operations could be implemented in at least two different ways: through the DIAL directive and through codes in the common lesson matrix.

The DIAL directive is an easy way of exchanging messages at the terminals. The players can send DIAL messages or they can be written into the lesson scenario. Having assigned terminal numbers to variable CALC names, targets of the DIAL message can be designated by using the name. The recipient of the message will also see the number of the sending terminal as a part of the message so that the player knows the identity of the one to whom he should address the reply.

Several applications are built around some form of trainee-directed iterative logic in which the lesson scenario consists of determining what the trainee is saying, doing it, and then returning to the same point in the scenario again for the next input. A wide variety of options may be available in this lesson loop. With this kind of logic it is also easy to regularly check one or more common lesson matrix entries, using specific code numbers to signify that the current trainee is being asked to do something by another member of the team. For example, a forward observer could be asked to supply data for particular sightings.

Although it would have been preferable to include a complete team training lesson scenario, constructing one is a non-trivial task and beyond the scope of the present effort. However, the examples are suggestive of some of the basic strategies that may be found useful.

### CONCLUSION

It is feasible to generate graphics from within PLANIT, given a useful set of language directives. It is also feasible to modify PLANIT such that team training lesson scenarios might be executed. Neither of these statements are much of a surprise nor do they constitute the outcome of this study that is really of interest.

Regarding graphics, it was important to define the nature of graphics which ought to be made available and for a reasonable cost. The blackboard model provided a useful frame of reference for this question since it is relatively easy to relate the expected benefits to be gained by the author by having such a facility at hand, and, since it is a familiar instrument, there is already an intuitive appraisal of its usefulness. That kind of capability can, in general, be made available to PLANIT authors with the exception of selective erase. However, magnification and reduction can be provided which goes beyond usual blackboard usage.

Regarding team training, it has been shown that PLANIT can mediate the kinds of lesson instructions which such a capability might need. This has been shown by

implementing a trial set of directives. What is not yet clear is the optimum set of directives which would facilitate the task of authoring such a lesson scenario. Other experience, which is abundant in the case of graphics, is sparse here.

Finally, the feasibility study was augmented to deliver more than just a report regarding feasibility. The PLANIT system was also modified by adding new language directives which permit both graphics and team training now. Although it is known that neither of these directives is optimum from the standpoint of authoring convenience, yet all are extremely flexible, permitting some sophisticated experimenting so that, if the follow-on development effort is undertaken to add the new capabilities, the Army will have already tested empirically the usefulness of the product they would be supporting. It is also possible that experimentation will reveal that the new capabilities are of marginal use, or that the recently added directives are sufficient. In either case, the approach used here which provided the tools for the experimentation will have saved a lot of time and money.

The Northwest Regional Educational Laboratory is pleased with the part they have played in probing these interesting new dimensions and looks forward to further assisting the Army Research Institute and other government agencies in refining these new capabilities to achieve optimum effectiveness.



## FOOTNOTES

1. The PLANIT Author's Guide, The Northwest Regional Educational Laboratory, Portland, Oregon, 1970  
  
The PLANIT Language Reference Manual, The Northwest Regional Educational Laboratory, Portland, Oregon, 1970
2. Hoyt, Butler and Bennik, Application of Tactical Data Systems for Training: Volume II - AI/DEVLOS Automation Studies, Research Memorandum 74-8, The U. S. Army Research Institute for the Behavioral and Social Sciences, July, 1974
3. Bennik, Hoyt and Butler, A CAI Course on Constructing PLANIT Lessons: Development, Content and Evaluation, Final Report, TM-5364/000/00, System Development Corporation, Santa Monica, California, July, 1974
4. Frye, TRANSL - The PLANIT Translator Program: Installation and Application, The Northwest Regional Educational Laboratory, Portland, Oregon, June, 1975
5. PLANIT Language Extensions Through Version 2.8, A Supplement to the PLANIT Language Reference Manual, The Northwest Regional Educational Laboratory, Portland, Oregon, March, 1976
6. Smith and Earn, MINING - Users' Manual (Machine Independent Numerical Interactive Graphics), Report No. 72-10, University of Colorado Computing Center, Boulder, Colorado, August, 1972
7. Meeker, Rogers, Madden and Miner, LIS/11--Laboratory Implementation System, Center for Computer-Based Behavioral Studies, University of California, Los Angeles, California, April, 1975
8. Vallee, Johansen and Spangler, The Computer Conference: An Altered State of Communication?, The Futurist, page 116, Vol. IX, No. 3, June, 1975  
  
Vallee, Network Conferencing, Datamation, page 85, May, 1974

## FOOTNOTES (cont.)

9. PLANET-1 User Guide, Institute for the Future,  
Menlo Park, California, August, 1975

APPENDIX A  
CONSULTANT REPORTS



DEPARTMENT OF THE ARMY  
ORGANIZATIONS AND SYSTEMS RESEARCH LABORATORY  
U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES  
1300 WILSON BOULEVARD  
ARLINGTON, VIRGINIA 22209

REPLY TO  
ATTENTION OF: PERI-OK

Dr. C. H. Frye  
Northwest Regional Educational Laboratory  
Lindsay Building  
710 S.W. Second Avenue  
Portland, Oregon 97204

Dear Chuck:

Unexpected time constraints (travel, budget cuts, etc.) are such that I had better put down my thoughts on our conference while I have the opportunity. The best way to proceed seems to be to address each of the questions you proposed in your original plan.

Question 1: I think we basically agreed here that there are two characteristics of general utility, i.e., functions/graphs and ability to create connected vectors. In both cases it appears the highest utility would be to permit student controlled modifications (such as the creation of functions, schematics, etc., but the use of "static" displays by an author was not precluded. As far as hardware, if it makes it any easier, we can probably consider the CRT as sort of a standard, in the abstract sense. Without a bit of empirical evidence, I submit that the CRT is likely to remain the most common, economical and popular computer driven display for some time. Also, as we tended to agree that the "blackboard" analogy was helpful I suggest this analogy continue as a sort of baseline for further discussion.

Question 2: The "new potential" would be that students could receive training in tasks which are by their nature, team tasks. The only advantage to employing a computer for such training is when the task requires that rapid calculations be made such that the "real world" appears to operate and change at a realistic rate, and/or when the task requires equipment which is either hazardous to operate or very costly-- in which case the computer can simulate the functions and output of such equipment (I use "equipment" in a generic sense). In effect then, it is such tasks that we should consider in determining the system





PERI-OK

Dr. C. H. Frye

requirements and hence the language modifications necessary to meet these requirements.

There is, of course, one interesting feature of interactions of the gaming nature: to predict and simulate all the variability likely to occur in human interactions is impractical, thus in those situations in which such behavioral variability has a major impact on goal achievement, interactive terminals should provide for direct communication links. As another aside, I tend to think the initial language modifications for inter-terminal training capability should emphasize cooperative rather than competitive situations. Most of our interest here is concerned with training personnel to work together efficiently rather than practice in competitive situations. For the time being it is easier to let the program carry the burden of acting as a competitor, besides giving the trainer greater control over the training problem.

Question 3: A good capability would be provided by this feature, but not really "equally good".

Question 4: I think this was sufficiently addressed in the conference. The major point was that one must distinguish between the specific trainee and his particular position/role in the team structure. Also, much of the LIS program suggests what must be included to achieve sufficient identification for communication. As for "idle time", I assume you mean what to do while communication is taking place. Clearly this is a time sharing problem, such that a particular station can continue working while messages are being received and transmitted.

Question 5: This involves being able to specify a set of conditions which must occur if a response is to be accepted. The problem of out of turn responses would not exist because the response would be rejected.

Question 6: Given you had good sequencing criteria when required, this problem would seem to be non-existent. That is, responses without conditionals attached would simply be acted upon in a time priority sequence, first come, first serve.

Question 7: Besides being aesthetically unsatisfying, such types of displays are clumsy to construct and, I suspect, relatively inefficient.

Question 8: I keep changing my mind on this one. Actually these alternative modes require such different language directives that you would probably need both eventually. I think in terms of the "Display Frame" we discussed, we are considering an enriched terminal. Application and integration with peripheral displays could probably be handled by a new set of directives within the current frames.

PERI-OK

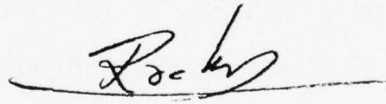
Dr. C. H. Frye

Question 9: I think we kind of implied that if one were constructing materials using one's own display, the assumption is these materials would be used with equivalent displays or not at all (Hence the notion of a "Display Frame" separate from the current frame types).

Question 10: The Tutor language and other languages which have provided graphics at a more or less meta-level can suggest directives. Currently there are no rules for defining simple, much less foolproof, directives.

Question 11: The answer is "Yes". The touch sensitive screen and a tablet type of input appear the most promising for future use, I might add that the Delphi study by the Annenberg School of Communication found that nearly everyone agreed better methods of inputting were needed than the traditional keyboard. However, I think, like the CRT, keyboards will be around for a long time and it wouldn't be all that restrictive to think keyboard.

Question 12: I believe this question was resolved. I think the first priority is to satisfy a specific user and then if things "work well" demonstrate features to other potential users. If they are desirable, formally make these part of the package.

  
LEON H. NAWROCKI  
Research Psychologist

Design Considerations for Adding Graphics and Multiple  
Terminal Capabilities to the Programming Language for Interactive  
Teaching (PLANIT).

Harry F. Silberman

With Consulting Assistance of Robert Meeker

University of California, Los Angeles

(For Northwest Regional Educational Laboratory)

This paper addresses three design considerations for adding graphics (non-alphanumeric processing) and multiple terminal (cooperative training) capabilities to the Programming Language for Interactive Teaching (PLANIT): (1) Functions to be served by such additions, (2) Problems to be solved, and (3) Lesson directives that might enable users to apply the new capability within the PLANIT language.

## I Graphics Capability

### A. Functions to be served

The specification of graphic processing requirements involves the cognitive processing demands and the physical characteristics of the embedding system to which the trainee must learn to respond. If the system requires the trainee to perceive complex functions or to visualize extraordinary spatial arrangements as is quite common in work on crystallography, astronomy, cytology, statistics and other applications in the physical sciences, graphic aids to conceptualization and cognitive processing in the trainee is of great help. Similarly, where the operational context,

within which the trainee must eventually perform, contains elaborate graphic displays or graphic response devices that cannot be easily simulated it is important to include such capabilities as a crucial and integral part of the training exercise.

Less important functions that may also be served by the addition of a graphic capability are the enhancement of trainee motivation and public relations. A dramatic demonstration of computer animation, for example, can be quite impressive. It is doubtful however that such pyrotechnics actually add very much to learning effectiveness; they may even contribute a distracting element. At current prices the cost of such enhancement is neither warranted nor recommended for PLANIT.

In the case of PLANIT a minimal set of graphic capabilities probably should be confined to drawing both straight and curved lines (including functions), less important, selective erasure, and, still less important perhaps, the ability to magnify or reduce those figures. Moving those figures, e.g., rotation, translation, is not considered necessary at this time. Such a capability would provide the user with a primitive "blackboard, chalk, and eraser." In the absence of specific target system hardware configurations the "blackboard" should suffice as a general graphic training vehicle. Where more fine grained, dynamic point-graphic capability is required, as in certain naturalistic environments, special nonportable versions of PLANIT would have to be developed for those unique systems on a case-by-case basis or, more likely, other means for training must be found. For example, if a particular tactical system featured a special analog joy-stick graphic control device and an esoteric display unit built for that system only, then training for use of those capabilities would best be provided after PLANIT instruction



through use of the operating system itself.

#### B. Problems

The most serious concern about adding graphics is its possible tradeoff with portability. PLANIT is presently best known for its portability and any serious compromise would make this language less valuable to the general civilian market. Some small changes to the current "MIOP" program that is used to couple PLANIT with unique parameters of individual machine configurations will probably be necessary but major changes to PLANIT would only complicate matters for regular users.

Unfortunately, the addition of new graphic capabilities to CAI languages carries with it a self-fulfilling prophecy. If the graphics capability is there it is likely to be used excessively and lessons that are developed will not be usable on machine systems lacking these graphics.

I recommend several precautions to offset the tendency of lesson designers to employ graphics for its own sake, and to maintain portability of PLANIT lessons.

1. Lesson designers might code their lessons in both graphic and nongraphic forms. For example, the nongraphic version of the lesson would contain associated material such as a Xerox copy of a map or graph while the graphics version would display such figures on the system terminal of the trainee. The requirement to prepare dual versions of lessons would insure portability in that users who don't have the graphic terminal equipment could still use the lessons. More important, the discipline of preparing the lesson in nongraphic form would reduce the tendency to employ graphics unnecessarily. Those applications of

graphic capability to provide static, large grain figures that are not critical to the lesson, that merely provide artificial enhancement, and don't make use of the dynamic graphic elements of the natural embedding environment, will be more detectable and likely to be dropped from production versions of lessons.

2. Include a by-pass feature in PLANIT so that sites without graphics can use lessons that contain graphics. This might be done by requiring all graphic lesson material to be constructed via a special graphics frame type, e.g., G. Regular users without graphics would not be permitted access to G-frame material but could operate normally all Q, M, D, P frames. When G-frames are confronted, auxiliary feedback messages might be generated on these users' terminals, e.g., "See graphics booklet page three," "map of Arizona appears about here," etc. Responsibility for preparing such feedback messages would reside with the original authors of the lessons using graphics. Users with graphic terminals would not receive these feedback messages. This feature could be accomplished by requiring the lesson author to declare "graphics only" (only users with graphic equipment may use this lesson) or to design feedback messages that provide the nongraphics user with a verbal by-pass (e.g., picture of weapon locations here) prior to every G-frame in lessons. Users would have to declare in "MIOP" the type of devices they have so that those without appropriate devices would be locked out from receiving G-frame material.

### C. Lesson design directives

The guiding principle for new directives by means of which authors may design lessons that use graphics is: KEEP IT SIMPLE. Assuming that the graphics capability will be implemented by the control of a "bucket of points" within PLANIT, several directives are suggested:

BLACKBOARD ON	Interprets and displays point matrix
BLACKBOARD OFF	Turns off this capability
ORIGIN x, y coordinates	Specifies location of origin for subsequent graphics
MOVE x, y coordinates	Moves pen without writing; moves to new position without plotting
(label) DRAW x, y coordinates (Calc Statement)	Connects new point with last point by line or by specified function
ERASE label, label	Removes lines or functions specified

These five directives should suffice for an initial "bare-bones" graphics capability. If a more elaborate set of control words is necessary I suggest three sources be consulted for candidates as new descriptors prior to launching ahead with the invention of new terms.

1. SIGGRAPH (ACM) has attempted to develop a standard set of graphic protocols. PLANIT interpretation of graphic directives ought to be consistent as possible with these agreements.
2. ARPA Net has also reached some consensus on graphic terms that should be consulted.

3. PLATO TUTOR Commands for graphics, drawing graphs, non-screen control, and special typing controls should be consulted. Inasmuch as PLATO system users have considerable experience with TUTOR and their graphic directives may have been evaluated and revised with empirical justification, it is a potential source of good ideas and we may ignore that experience only at the cost of repeating their history of problems in graphic lesson construction.

## II. Multiple Terminal Capability

### A. Functions to be served.

A cooperative training mode provides a dramatic and realistic demonstration of the consequences of poor teamwork (which often serves to motivate subsequent learning), and direct training in coordination skills (communication, cooperation, joint timing, decision making, priority setting). Although individual training can provide these functions without loss of stimulus control by simulating the embedding social environment it is far cheaper to use the cooperative mode. In individual training the lesson author must anticipate all the interactions, errors, and instabilities normal to the system environment, e.g., indecisiveness of supervisors, poor timing in subordinate responsiveness to command, and each position must be trained separately. In cooperative training greater realism is possible because people "play" their own positions and the cost of expensive simulation of those positions is avoided. Furthermore the involvement of the total tactical team in a simultaneous



system exercise furnishes a more potent demonstration of the consequences of poor team performance ("Because you didn't coordinate actions, we lost San Francisco and Portland.").

I am convinced that the training potential of PLANIT would be greatly improved by the addition of a teaming capability. Lessons could be designed to progressively escalate the interactive demands on members of any kind of organization, stressing crucial communication functions until the many social skills were well established for smooth system performance.

#### B. Problems

The most serious concern about adding a multiple terminal capability to PLANIT is the possibility that the complexities of designing a cooperative exercise may be prohibitive and only extremely competent authors could successfully construct more than the simplest of exercises. This is not a trivial concern and it will not be known whether the problem can be dealt with until a working prototype is available to "average" designers for trial lesson development. The additions to PLANIT must feature very simple procedure control directives. It will be a delicate piece of surgery to expand PLANIT to incorporate new directives that don't compromise its user-orientation nor its present individual training applications. A good argument could be made for constructing an entirely new language such as "LIS," developed by CCBS at UCLA, with ARPA support. But separate maintenance costs and portability requirements lead me to recommend incorporation into a comprehensive PLANIT of a few of the most critical directives from LIS. A single comprehensive author language with capacity for both individual and group training, with the portability (machine independence) of PLANIT, plus a graphic capability has enormous

potential for advancing the state of the CAI art.

c. Lesson design requirements

Most of the lesson design features that are unique to the teaming capability involve some kind of communication among workers in an organization or tactical system. Such communication is most often work-oriented. For example, the operational system design places certain demands on working arrangements; the technology also imposes itself on the nature of the task. Communications may also be non-work-oriented. For example, much communication in any organization concerns the less formal social interactions that define the "style" of a given working group. Both work and non-work communication occurs in verbal and nonverbal modes. The four cells of the following table summarize these alternative forms of communication.

	Work oriented	Non-Work-Oriented
Verbal	(1)	(3)
Nonverbal	(2)	(4)

- (1) Questions, answers, statements of opinion, instructions, suggestions, and other work-related messages transmitted face-to-face or by other media, e.g., phone, notes, computer display
- (2) Actions of one or more people which affect the work of one or more other persons, e.g., rate busting, slacking off, cooperation, competition, accidents, absenteeism, switch actions, timing, errors, manners.
- (3) Personal talk, joking, gossip, breaks, largely face-to-face transactions but notes and phone also used.
- (4) Facial expressions, body language, appearance, dress, distance etc.

Some forms of team training attempt to control all four of these modes, e.g., management training exercises. But PLANIT exercises cannot deal readily with category 3 and 4 and should be confined to category 1 and 2 in a manner similar to LIS.

#### D. Lesson Design Directives

The first set of directives for the PLANIT teaming modification is to incorporate a means of identification of participants; a second set is to make provision for verbal work-oriented communication; and finally, an approach must be described for implementing these new directives within the PLANIT lesson logic. Each of these items are discussed below.

##### 1. Identification

At least three to five hierarchical levels are needed to distinguish among individuals and groups. Messages must be capable of being addressed to everyone (ALL), to people in a particular functional role (ROLE) to a particular unit, section, or subdivision of that role (UNIT), to a particular shift, replication, or working team (SHIFT), and to a specific person (NAME). For any given organization these identifications will be differently indexed. In a field artillery observation battalion, ROLES may include Fire Direction Center, Forward Observers, Fire Batteries; A UNIT may include radar operations or a particular fire battery; a SHIFT may be the swing shift or the graveyard shift. A NAME might be John Smith, weapons assignment officer. In a school, the Principal (ROLE) may ask the Biology classes (UNIT) in the evening session (SHIFT) to assign Mr. Johnson (NAME) to room 45.

Protocol for specifying identification may follow the existing PLANIT convention for identification of editing targets (frame, group, line). The identification descriptors would follow the form

ID	Identification indicator
role, ID	ROLE identification
role, unit, ID	UNIT identification
role, unit, shift, ID	SHIFT identification
role, unit, shift, name, ID	NAME identification

In some cases the prefix ALL can be used to cross index members of a group. For example:

ALL, shift, ID	All members of a SHIFT across all units and roles are designated
----------------	--

Where individuals are members of more than one hierarchical organization as in a matrix form of organization, they may be identified with both, and actions or messages directed to them in either organizational line will reach them.

When individuals are specified, messages or actions will only apply to those individuals, but where groups are designated as in SHIFT, UNIT, and ROLE, the messages or actions will apply to all members of those categories.

## 2. Verbal communication

Two forms of verbal communication are recommended for PLANIT: VOTING and MESSAGES.

### a. Voting

In many exercises it is necessary to poll all members of a group or subgroup; their responses must be collected



and tallied; the result of the poll compared against some decision criterion, and consequent action taken. A convenient descriptor is used in LIS to accomplish these procedures:

VOTE voteridentification, (majority, unanimous) sets up a no-wait-read voting procedure for all persons in a designated group, holding all members in the set until all have completed their vote.

HOLD identification Inactivates processing of group members until all are in a certain frame.

b. Messages

Control directives for message transmission are also specified in LIS, though descriptors might be altered for author convenience as follows:

SEND recipientidentification(s) Permission to send regular messages

SEND PRIORITY recipientidentification(s) Permission to send priority messages

SEND CENSORED recipientidentification(s) Permission to send priority messages to someone through a reviewing monitor who can choose to suppress, release, or alter the message enroute

RECEIVE LIST Searches through person's in-basket and presents list of message numbers and originators (envelope list)

RECEIVE READ    Permits person to select a message  
                  from the list for display (open  
                  letter and read)

RECEIVE PRIORITY    Displays high priority messages

3.    Nonverbal communication

Two categories of nonverbal communication would be important to a teaming capability: consequences of other person's actions and procedure control. These are also found in LIS.

a.    Consequences

Actions, options or choices made by one player may affect other players. This is a feature of any interactive group. For example in the commons dilemma game short-term greed by individuals who claim common resources have long-term negative consequences for all. PLANIT must be able to specify such consequences. Whenever a person's actions are expected to have an impact on others a "D" frame might be used to accomplish the intended effects by specifying the conditions on which actions are to be taken in other person's lesson segments. For example, assume a directive, IMPACT, is turned on indicating that henceforth actions will affect other stations as well as this one. The affected stations must be designated and the conditional statements included as part of the lessons of the affected parties:

SET IMPACT recipientidentification Action  
                  will affect the recipient parties,  
                  (sets counter for each of those  
                  recipients)

IF IMPACT (relational) (number) (action)      In lesson  
segment of recipients, "D" frames are  
used to detect impact of other people's  
actions and specify intended consequences  
for recipient

b. Procedure control

A modified PLANIT should also include the ability to  
change legality of interactions. For example, if a  
commander decides that a given operator should be granted  
permission to use a shortcut in a communication sequence  
he should be able to do it. This might be done as follows:

SHORTCUT ON      When players are responding to repetitive  
queries, they may be permitted to  
stack responses so that responses may  
be evaluated without the intervening  
displays, thus short circuiting  
routine step-by-step solicitations of  
student responses in a sequence that  
is quickly learned by alert players  
who anticipate subsequent action  
requests.

SHORTCUT OFF      Turns shortcut off.

Other unspecified procedure control directives will be  
needed for the asynchronous mode of operation as when a  
command decision alters the organization, personnel  
composition, or message routing procedures for a system  
in the course of an exercise.

#### 4. Lesson design logic

Assuming that team lessons in PLANIT will be controlled both by a series of frames and by a common data base containing numeric information, the control of the architecture of team exercises might best retain the old PLANIT lesson design strategy by allocating ranges of frames to the various identification categories:

Set up information correlating ID with ranges of frame numbers

frame 1 ID=x	(Each lesson segment would contain a series of frames that specify the interactions for a particular role, unit, shift, name ID)
frame n ID=y	
frame n+x ID=z	(Each lesson segment would communicate with a common data base and would also contain separate buffer space allocations for actions unique to that segment.)
End of Exercise	

For example, the x segment of the lesson would provide the instructional sequence for one role, the y segment would serve a different role, the z segment another. Setting impact variables in one segment and sensing those variables in another segment would implement nonverbal interaction. Using the various message directives in any segment would implement verbal communication among players. These directives would appear in groups 3 and 4 of the Q and M frames within those segments where player interaction is to occur.

PLANIT author manuals would have to be upgraded not only to include interactive functions but also to augment the definition and conceptualization of existing functions (e.g., Questions-displays may also be thought of as Option-displays, and, in association responses may be thought of as Actions).



APPENDIX B

CONSULTANT CONFERENCE AGENDA

## PLANIT WORKING SESSION

September 22-23, 1975

### I. Desired entry level proficiencies.

It is hoped that the participants will use the two-day preparation period to thoroughly acquaint themselves with the PLANIT conventions for the following:

1. Meanings of each of the four groups in the question/multiple choice frames.
2. Presenting information to the trainee via
  - . Group 2 of the question/multiple choice frames
  - . The action commands (F, R and C)
  - . The Calc PRINT and ALIGN commands
3. Matching answers in group 3 of the question/multiple choice frames
4. Calc arrays (declared by the MATRIX command)
5. Samples of control directives in Calc (e.g. PHONETIC) and how and where they can be manipulated in the lesson (e.g. after a zero tag in group 3, and wherever a C: action command is legal)

In addition, each participant should review, in relation to his own past experiences, ways in which people have teamed in a man/machine environment to solve a common problem, how computer-generated graphics have aided this process, and to what extent it was important that the trainee could respond by pointing at (or moving a pointer along) the CRT screen as opposed to responding on a typewriter keyboard.

### II. Goals for the working sessions.

The goal is to assess needs and to recommend specific user-level modifications to the PLANIT language.

Relative to assessing needs, the questions in section III below should be explored in terms of the kinds of training which cannot now be done with PLANIT (or other CAI systems) because it lacks specific capabilities, and where these needs fit on some priority scheme.

Relative to the language recommendations, the most desirable and least disruptive modifications would be those of the type which fit the current structure of the PLANIT language syntax and would not destroy the compatibility which now exists among the various PLANIT installations. Adding to the current language would be less disruptive than changing it. However changes are not to be completely ruled out.

### III. Some questions to be addressed in the working sessions.

1. What kind (quality, characteristics, etc.) of computer displayed graphics of the type that require additional PLANIT interfacing (e.g. vector generating CRT, plotter, etc.) would be of significant value in training?
2. What new training potential could be expected from a teaming capability in PLANIT such that a single lesson scenario would interact simultaneously with several trainees (or game players)?
3. Could an equally good teaming capability be provided more simply by enabling trainees to share data in some automatic fashion while interacting with their own particular lesson scenario?
4. How would an author identify each of the participants on his lesson and how would he designate to whom each message is being sent, from whom each is being received, and what to do with idle time?
5. If a sequencing of responses is implied by a training strategy, how would that sequence be enforced or what could be done with the response from a participant who submitted it out-of-turn?
6. If a random pattern of responding is appropriate (rather than sequenced), how will the author cause his lesson to react to a response without the undue delay of taking each in turn?
7. Recognizing that PLANIT can now display any graphics of the type that can be created on a

regular typewriter and displayed on a CRT screen (like Snoopy pictures for example), what additional graphic-producing directives should be available to authors?

8. Should the enriched graphic screen be the display part of the trainee terminal, should it be only an adjunct to a conventional terminal that has other display provisions, or should it optionally work either way?
9. What provision should be made, if any, for the display of graphic materials on devices which differ from that which the author used while preparing the materials? Should any new graphic capabilities be limited to those features which one can expect from most graphic instruments?
10. What are some simple, foolproof, machine (device) independent directives which can be added to the PLANIT language to implement the above. They should be human engineered to let the author specify what he wants in a one-to-one relationship with the function he wishes performed without the worry of timing, screen calibrations, device differences, other users on the system, etc., etc.
11. Would the addition of more sophisticated graphics stimulate the need for new response mode alternatives (such as light pens, touch sensitive screens, pointing with a cursor, etc.)? If so, consider the PLANIT language needs for these devices in the terms of questions 9 and 10 above.
12. How extensive a trial should be given to any of the above additions before that feature is advertised as being a part of the "released" PLANIT system?

#### IV. Outcomes.

Each participant is being asked to prepare in writing his observations and recommendations concerning the agenda items in the two-day working session, including the above 12 points and others that may be added. The recommendations are not expected to be complete solutions but rather to provide guidelines for the remainder of the feasibility study of which this session is a part. Do as much as two days will allow.



V. Arrangements.

As your own situation will allow, you will be paid daily consulting rates for:

- . Two days preparation prior to the working session
- . Two days of the working session
- . Two days following the working session for preparing your recommendations.

The working session will convene in Portland, Oregon on September 22 and 23. You may spend these two or as many as six days in Portland at a reimbursement rate of \$25 per night. If you choose Portland for the preparation period, access to PLANIT will be provided after 5 PM. If you choose to remain in Portland to prepare your recommendations, a typist will be available. Other reimbursements will include Coach air fare and terminal transportation. A rental car will not be necessary for the work in Portland especially if you stay at Riverside West Motel which is less than a block from this Laboratory. A DART bus will bring you in from the airport in 20 minutes to a point about eight blocks from the motel. Chuck Frye will help you with any arrangements you wish.

APPENDIX C

PRINTER GRAPHIC

